

# 국제 표준 기반 열차제어시스템 소프트웨어의 정적 테스팅 지원도구의 개발

논문
58P-2-3

## Development of Static Testing Tool Related Int'l Standard for Railway Signaling Software

황 종 규<sup>†</sup> · 조 현 정<sup>\*</sup>  
(Jong-Gyu Hwang · Hyun-Jeong Jo)

**Abstract** - Recent advances in computer technology have brought more dependence on software to train control systems. Hence, the safety assurance of the vital software running on the railway system is very critical task and yet, not many works have been done. While much efforts have been reported to improve electronic hardware's safety, not so much systematic approaches to evaluate software's safety, especially for the vital software running on board train controllers. In this paper, we have developed the static software testing tool for railway signaling, especially Fagan Inspection module and checklists supporting module. This static testing tool for railway signaling can be utilized at the assessment phase, and also usefully at the software development stage also. It is anticipated that it will be greatly helpful for the evaluation on the software for railway signalling system.

**Key Words** : Fagan Inspection, Software Checklist, Software Static Testing, Railway Signalling System

### 1. 서 론

열차제어시스템은 고속으로 동작하는 철도차량의 속도 및 진로를 실시간적으로 제어해야 하는 철도시스템에서의 핵심적인 장치이다. 이러한 열차제어시스템은 최근 기존의 기계 및 전기적인 장치로부터 컴퓨터시스템으로 전환되고 있으며, 특히 안정화된 하드웨어 보다는 소프트웨어에 그 의존성이 급격하게 증가하고 있어, 소프트웨어의 의존성이 급격하게 증가하고 있다. 최근의 컴퓨터 기술의 발달에 따라 지능화 및 자동화를 위해 열차제어시스템의 소프트웨어가 더욱 복잡해지게 되면서, 시스템에서 소프트웨어가 차지하는 비중이 더욱 증대되고 있다. 이러한 철도시스템 소프트웨어의 주요한 특징은 실시간성과 고신뢰도, 그리고 안전성을 들 수 있다. 즉, 철도시스템 소프트웨어가 고신뢰도와 안전성을 만족하지 못하는 경우 열차의 충돌 등 사고를 발생시킬 수 있는 바이탈한 시스템이다. 따라서 이러한 철도시스템에 임베디드화 되는 제어 소프트웨어의 안전성을 검증하는 것이 중요한 문제로 대두되기 시작했다.

이에 따라 철도시스템 소프트웨어 안전성관련 기준들은 IEC 61508-3[1]과 IEC 62279[2]에 의해 국제표준화 되었고, 또한 국내에서도 철도안전법[3]이 제정되는 등 국제표준에서 요구하는 각종 테스팅 및 검증활동을 요구하기 시작했다. 철

도시스템 중 특히 바이탈 제어장치인 열차제어시스템의 소프트웨어 안전성 평가를 위해서는 관련된 국제규격에서 소프트웨어의 분석 및 측정을 통한 정량적인 평가뿐만 아니라 소프트웨어 개발과정의 안전성 활동에 대한 문서 등의 검증을 통한 정성적인 분석도 안전성 평가의 중요한 요소로 요구하고 있다.

특히 대부분 안전무결성 등급(SIL : Safety Integrity Level) 등급이 3 또는 4로 분류되는 ATP나 전자연동장치와 같은 바이탈한 열차제어시스템 소프트웨어의 경우 Fagan Inspection을 정적분석의 방법으로 'HR : Highly Recommend' 조건으로, 기능 안전성 평가를 위한 방법으로 체크리스트 테스팅을 'HR' 조건으로 규정하고 있다. 이러한 Fagan Inspection과 체크리스트 관리는 평가 자체가 아니라 평가 과정에 대한 품질확보를 위한 측면이 강한 방법이지만 국제표준에서 정적 테스팅을 위한 방법의 하나로 제시하고 있는 등 정적 테스팅이 열차제어시스템 소프트웨어 안전성 평가에 있어서 매우 중요한 필수적인 부분이다[1][2].

하지만 국내에서는 철도시스템 소프트웨어의 정적 테스팅을 위한 두 가지 방법을 지원하는 도구가 개발된 적이 없고 Fagan Inspection의 경우 몇몇 외국 제품들에 의해 상용화되고 있다. 본 논문에서는 이러한 철도소프트웨어 안전성 평가체계 구축의 일환으로 국제규격에서 철도소프트웨어의 안전성 평가를 위해 필수적으로 요구하고 있는 체크리스트와 Fagan Inspection을 지원하기 위한 도구를 개발하였다. 본 논문은 2장에서는 열차제어시스템 소프트웨어 안전성 평가 개요를, 3장은 소프트웨어 Fagan Inspection 분석 모듈의 설계 내용, 4장은 안전성 평가 체크리스트 관리도구의 설계내용, 5장을 정적 테스팅 도구의 구현결과를 설명하고 마지막으로 6장에서 결론을 설명하는 것으로 구성한다.

<sup>†</sup> 교신저자, 정회원 : 한국철도기술연구원 선임연구원 · 공박

E-mail: jghwang@krri.re.kr

\* 정회원 : 한국철도기술연구원 주임연구원

접수일자 : 2009년 2월 10일

최종완료 : 2009년 3월 24일

## 2. 열차제어시스템 소프트웨어 안전성 평가

열차제어시스템 임베디드 소프트웨어는 개발초기부터 테스팅 과정을 통해 버그를 확인하여 품질비용을 낮출 수 있으며, 또한 개발 완료 후 검증과정에서도 안전성 평가 과정이 필수적으로 요구되고 있다. 열차제어시스템 소프트웨어는 하드웨어에 의존적이고 높은 안전성이 요구되는 바이탈 소프트웨어여서 다른 어느 제어시스템 소프트웨어보다 엄격한 안전성 평가가 요구되고 있으나 아직 이의 지원을 위한 도구들에 대한 연구가 많이 진행되고 있지 않고 있으며, 단지 산업용 임베디드 시스템의 소프트웨어를 대상으로 일부 항목에 대한 도구들이 사용되고 있다.

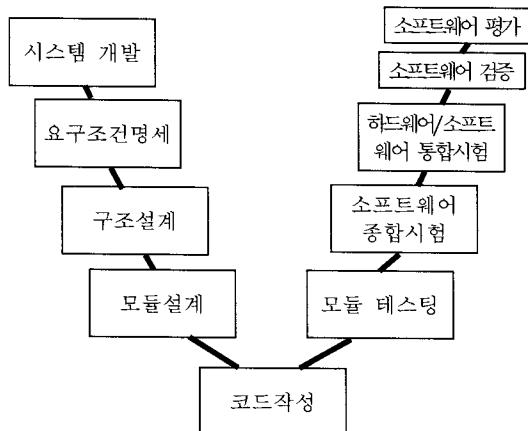


그림 1 소프트웨어 개발 수명주기  
Fig. 1 Software Development Lifecycle

본 논문에서의 열차제어시스템 소프트웨어 안전성 평가 기술 개발 및 체계 구축을 위해 필수적으로 요구되는 정적 테스팅을 위한 Fagan Inspection과 체크리스트 관리를 위한 도구를 개발하였다. 그림 1은 IEC 62279에서 제시한 소프트웨어 수명주기를 나타낸 것으로 일반적인 제어시스템 수명주기와 유사하다. 소프트웨어의 안전성 평가를 이 수명주기의 마지막 단계인 ‘소프트웨어 평가’ 단계에서 동적 테스팅 위주로 수행하지만, 철도의 열차제어시스템과 같은 바이탈 소프트웨어는 동적 테스팅에 더불어 수명주기 전 단계에서 걸쳐서 수행하는 프로세스 리뷰 등과 같은 정적인 평가방법을 매우 중요한 평가요소로 요구하고 있다. 이러한 바이탈한 소프트웨어의 안전성 평가를 위한 정적 분석 방법 중 본 논문에서는 정형화된 인스펙션 방법의 하나인 Fagan Inspection과 체크리스트를 지원하기 위한 도구를 웹기반으로 구현하였다.

Fagan Inspection 방법은 소프트웨어의 안전성 평가를 위한 정적 테스팅 방법의 하나로 철도관련 국제표준에서 ‘HR’로 요구하고 있는 방법이다. 이는 소프트웨어 개발의 모든 단계에서 정의된 규칙을 기반으로 인스펙션을 수행하고 결과를 등록 및 보고하는 소프트웨어의 품질 및 안전성을 확보하기 위한 방법이다. Fagan Inspection은 소프트웨어의 리뷰 과정을 공식적인 절차에 의해 하도록 하는 페이인에 의해 제안된 방법으로 인스펙션 과정을 통해 코드의 70~80%의 결함이 발견된 것으로 소개하고 있다[4][5]. 이러한 리뷰 절차를 체계적으로 수행하기 위한 Fagan Inspection을 소프

트웨어 개발 과정에서 적절하게 수행했는지를 확인하는 것이 소프트웨어의 정성적 안전성 평가에서 매우 중요한 요소이며, 이러한 인스펙션 프로세스가 자동화된 도구에 의해 지원된다면 보다 효율적으로 수행될 수 있을 것이다.

체크리스트는 철도소프트웨어의 기능 안전성 평가를 위해 국제표준에서 요구하고 있는 방법으로, 이 또한 소프트웨어 개발의 모든 단계를 통해 시스템의 기능 안전성을 평가하는 것이다. 체크리스트는 대부분 각 시스템의 적합성을 평가하기 위한 항목들로서, 결과적으로 체크리스트의 질문을 사용하여 시스템이 반드시 해야 하는 사항과 연관이 없음이 확인되어야 한다. 본 논문에서는 철도 소프트웨어 관련 국제표준인 IEC 61508-3과 IEC 62279의 분석을 통해 열차제어시스템 소프트웨어 안전성 평가 가이드라인을 도출하였고, 이를 정적 테스팅을 위한 체크리스트로 활용하였다.

Fagan Inspection과 체크리스트 방법 모두 정적 안전성 평가방법으로 동적 테스팅을 통해 발견하기 어려운 결함, 개발 중간 산출물의 누락, 준수하여야 할 표준 위반, 요구사항 결함, 개발설계 결함 등을 확인할 수 있어 열차제어시스템과 같은 높은 안전성이 요구되는 소프트웨어의 안전성 평가에서는 동적 테스팅을 통한 정량적 평가에 더하여 필수적으로 요구하고 있는 방법이다[6][7].

## 3. Fagan Inspection 모듈

### 3.1 Fagan Inspection 모듈

Fagan Inspection 방법은 철도 소프트웨어관련 국제 표준에서 소프트웨어의 안전성 평가를 위한 정적 테스팅의 하나로 요구하고 있는 방법이다. 이는 소프트웨어 개발의 모든 단계에서 정의된 규칙을 기반으로 인스펙션을 수행하고 결과를 등록 및 보고하는 소프트웨어의 품질 및 안전성을 확보하기 위한 방법으로, 그림 2와 같은 인스펙션 수행절차를 통해 소프트웨어 개발과정에서의 요구사항이나 설계의 적정성 및 잠재되어 있는 오류를 확인할 수 있다. 안전성 평가를 위해서는 소프트웨어의 각 개발단계에서 이러한 인스펙션이 정의된 규칙 및 절차에 따라 수행되었는지와 인스펙션 결과의 처리여부 등의 평가를 수행하게 된다.

이러한 정적분석 방법의 하나인 인스펙션은 표1과 같은 7 가지의 단계를 가진다. 소프트웨어 개발의 각 단계별로 각각

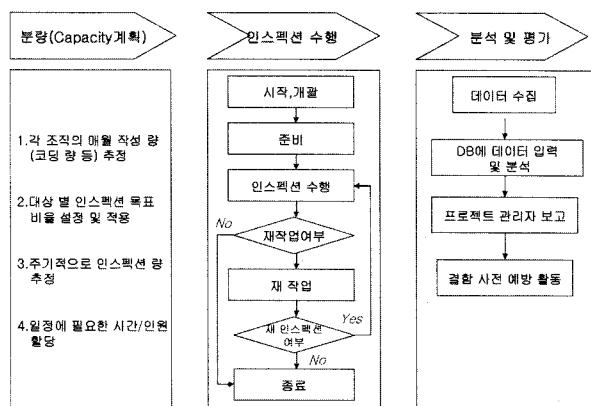


그림 2 인스펙션 절차

Fig. 2 Fagan Inspection Procedure

표 1 Fagan Inspection의 단계  
Table 1 The Phase of Fagan Inspection

단계	설명
Planning	<ul style="list-style-type: none"> <li>Inspection 할 산출물이 entry criteria를 만족함을 확인</li> <li>Inspector 역할 할당 (Moderator, Author, Reader, Tester)</li> </ul>
Overview	<ul style="list-style-type: none"> <li>Inspection 팀이 preparation 단계를 잘 실행할 수 있도록 배경 설명, context, rationale에 대한 교육 실시</li> </ul>
Preparation	<ul style="list-style-type: none"> <li>각자 Inspection 할 산출물의 습득</li> <li>각자 맡겨진 역할을 수행할 수 있도록 준비</li> <li>Defect로 단정 짓지 말고 Inspection 회의 시 질문 사항 기록</li> </ul>
Inspection	<ul style="list-style-type: none"> <li>Find Defect (defect 해결책 또는 개선책을 거론하지 말 것)</li> </ul>
Process Improvement	<ul style="list-style-type: none"> <li>향후 defect 발견을 항상 시킬 수 있도록 실행 단계 검토</li> <li>Systematic defect 식별과 Recommended fixes</li> </ul>
Rework	<ul style="list-style-type: none"> <li>모든 발견된 defects 수정 및 조사항목 보완</li> </ul>
Verification	<ul style="list-style-type: none"> <li>모든 보완 사항과 해결책이 적합한가를 검증</li> </ul>

표 1과 같은 인스펙션을 수행하고 또한 그 과정을 기록하여야 안전성 평가 시에 평가될 수 있도록 하여야 한다. 이러한 Fagan Inspection은 지금까지는 대부분 오프라인 방식에 의해 수행되어 왔다. 하지만 열차제어시스템과 같은 높은 안전성이 요구되는 바이탈 소프트웨어의 경우 이러한 인스펙션을 필수적으로 요구하고 있고 또한 그 결과가 안전성 평가에서 검증되도록 규정되어 있어, 지금까지와 같은 오프라인 방법에 의해 수행할 경우 인스펙션 과정이 비효율적일 뿐 아니라 안전성 평가를 위한 문서화 작업이 불가능하여 이를 효과적으로 지원할 수 있는 도구의 개발이 필요하다.

### 3.2 Fagan Inspection 모듈의 설계

국제표준에 의한 열차제어시스템 소프트웨어 테스팅 자동화 도구의 한 부분으로써 프로젝트 단위로 작업을 구분하여 소프트웨어 개발 절차에 따라 각 단계 별 Fagan Inspection 모듈을 이용하여 개발하는 소프트웨어의 안정성을 평가하는 웹기반 테스팅 도구를 설계하였다. 본 논문에서는 C나 C++에 적합한 Fagan Inspection을 위한 규칙을 정의하였고, 정의된 규칙을 기반으로 인스펙션을 수행하고, 인스펙션 수행 결과 등록 및 관리·보고서 작성 기능, 단계별 산출물을 등록, 시험대상 코드 등록, 검사 수행 및 결과 등록, 검사 결과 보고 및 조회 등의 기능을 구현하여 안전성 평가를 지원하기 위한 Fagan Inspection 도구를 설계하였다.

본 논문에서는 그림 2와 같은 인스펙션 절차를 일반화하여 그림 3과 같은 기본 단계를 바탕으로 열차제어시스템 소프트웨어 개발 수명주기 전반에 걸쳐 적용 가능하도록 하였다. 또한 열차제어시스템의 각 소프트웨어, 혹은 각 모듈 별로 구성되는 프로젝트 단위로 Fagan Inspection을 적용 가능하도록 하는 기능이 필요하다. 이에 따라 각 프로젝트 별로 관리 및 참여자(User)의 관리 역시 함께 필요하다. 즉, 정의된 규칙을 기반으로 Fagan Inspection 수행하고 그 결과를 등록 및 보고할 수 있도록 하여 단계 별 산출물을 등록하여 Fagan Inspection에서 이용이 가능하도록 지원하고 시험대상 코드 역시 웹 시스템에 등록하여 Fagan Inspection에 참여하는 참여자들이 함께 정보를 공유하고, 열람하며, 그에 따른 검사를 수행할 수 있도록 설계하였다. 각 단계별 인스펙션의 결과를

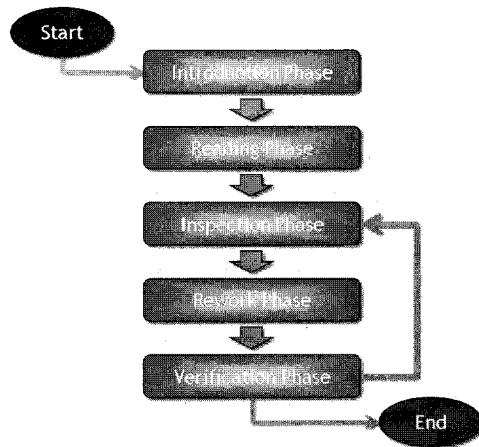


그림 3 설계한 Fagan Inspection 단계

Fig. 3 Designed Phases of Fagan Inspection

프로젝트 관리자가 획득하여 활용할 수 있도록 웹 시스템에 등록할 수 있도록 하고, 또한 인스펙션의 결과들을 취합하여 자동으로 보고서가 작성될 수 있도록 설계하였다.

또한 Fagan Inspection 모듈은 크게 관리자와 일반 유저 관점으로 구분하여 제한된 기능이 활용될 수 있도록 하였다. 관리자 관점에서는 이반 유저의 등록이나 수정, 삭제, 검색 등의 일반 유저 관리기능과 프로젝트 등록, 수정, 삭제, 프로젝트 관리자 설정, 검색 등의 프로젝트 관리 기능을 구성하였다. 일반 유저 관점에서는 개인 정보의 관리, 참여 프로젝트 리스트 및 프로젝트 정보 열람 등의 기능으로 구성하였다. 일반 유저 관점에서 Fagan Inspection에 대한 기능은 표 2와 같이 다시 Fagan Inspection에 참여하는 참여자의 종류별로 기능을 설계하였다.

표 2 인스펙션 참여자와 주요 역할

Table 2 The Inspectors and their main Roles

참여자의 종류	설명
프로젝트 관리자	관리자가 정의한 프로젝트의 관리자로써 프로젝트의 전반적인 관리를 담당한다. 프로젝트에 참여해야 하는 참여자를 설정할 수 있다.
Inspection Author	프로젝트 내에 각 단계 별 Inspection의 개설 역할을 담당한다. Inspection에 참여하는 참여자를 설정하고 각 참여자 별 권한을 설정, 수정할 수 있다. 또한 Inspection을 진행하는데 필요한 초기 자료 수집 및 관련 자료들을 Web System에 등록을 해야 하는 의무가 있다.
Inspection Moderator	프로젝트 내에 각 단계 별 Inspection에서 실제 Off-line 토의 시 토의 종재자 역할을 담당한다. 토의가 원활하게 진행 될 수 있도록 토의의 시작을 알리며 토의 중간에 참여자들 간의 공격적인 발언을 자제시키며 토의의 마무리를 이끌어내야 하며 다음 토의의 계획을 세워 다른 참여자들에게 알려줘야 하는 의무가 있다.
Inspection Reviewer	프로젝트 내에 각 단계 별 Inspection에서 Inspection Author가 등록한 관련 자료들을 Web System을 통해 전달 받아 그 자료들을 검토하여 그 검토 의견을 등록하는 역할을 담당한다.
Inspection Reader	프로젝트 내에 각 단계 별 Inspection에서 실제 Off-line 토의 중 Reviewer들에게 Inspection 대상에 대한 설명을 함으로써 Reviewer들의 Inspection 대상에 대한 이해도를 높여주는 역할을 담당한다. 그리고 Inspection을 통해 밝혀진 오류를 수정하는 작업을 담당한다.

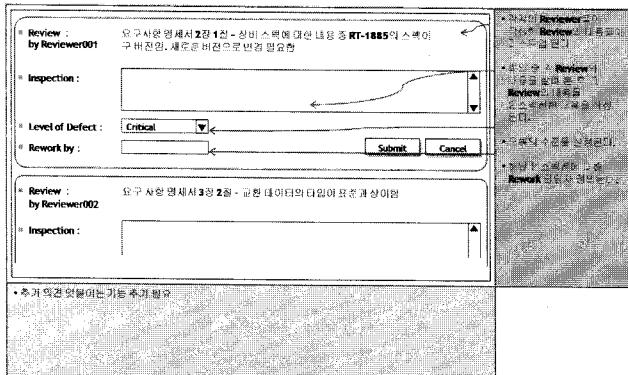


그림 4 인스펙션 단계의 화면 설계

Fig. 4 Designed Windows for Inspection Phase

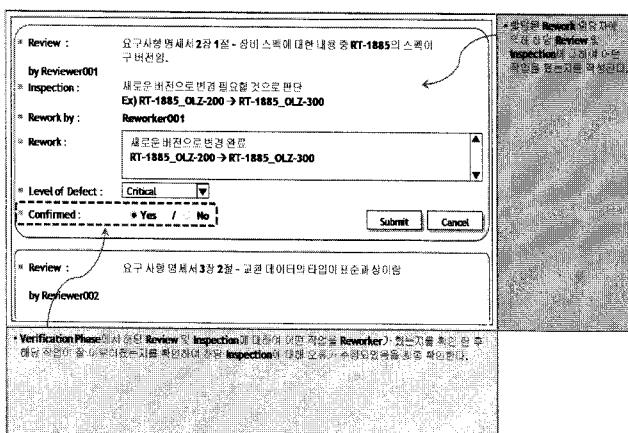


그림 5 Rework &amp; Verification 단계의 화면 설계

Fig. 5 Designed Windows for Rework &amp; Verification Phase

그림 4와 5는 앞에서 설명한 바와 같은 Fagan Inspection의 단계들 중 'Inspection Phase'와 'Rework Phase & Verification Phase'을 위해 설계한 윈도우를 나타낸 것이다. 그림 4와 같은 웹기반의 윈도우를 통해 각 리뷰어들이 작성한 내용을 살펴본 후 그 내용을 인스펙션한 내용을 작성하게 되며, 그림 5는 인스펙션한 결과를 해당 리뷰어가 확인하고 그에 응답하고 또한 최종적으로 해당 인스펙션이 반영되었는지를 확인하는 윈도우이다.

#### 4. 체크리스트 모듈

##### 4.1 소프트웨어 안전성 평가 가이드라인

국제표준에서 열차제어시스템의 기능 안전성 평가를 위한 방법의 하나로 체크리스트에 의한 관리를 요구하고 있다. 체크리스트는 대부분 각 시스템의 적합성을 평가하기 위한 사항들로서, 결과적으로 체크리스트의 질문을 사용하여 시스템이 반드시 피해야 하는 사항과 연관이 없음이 확인되어야 한다. 본 논문에서는 3장에서 설명하였듯이 철도소프트웨어 안전 관련 국제표준을 통해 열차제어시스템 소프트웨어 안전성 평가 가이드라인을 도출하였으며, 이 평가 가이드라인의 주요 내용은 다음과 같다[8].

소프트웨어 안전성 평가는 소프트웨어가 설계과정에서 도출된 목표 안전도 등급을 어느 정도 달성하였는가를 평가하

고, 그 결과에 의거하여 목표의 달성을 여부를 판단하는 과정이다. 소프트웨어 안전 무결성 수준(SWSIL : Software Safety Integrity Level)은 소프트웨어 자체적으로 정의되지 않으며, 소프트웨어를 적용하는 시스템의 SIL과 동일하도록 결정된다. 그러나 소프트웨어의 오류가 시스템으로 전파되지 않도록 할 수 있는 경우에는 이보다 낮은 수준으로 정할 수 있도록 하였다.

소프트웨어 개발 각 단계별 만족해야 하는 요구사항들이 국제표준에 제시되어 있으며, 이를 요구사항들로부터 작성한 가이드라인에서 체크리스트를 도출하였다. 정의하고 있는 요구사항은 크게 문서화, 소프트웨어 품질관리 체계, 소프트웨어 안전수명주기 요구조건, 기능안전성 평가의 네 가지 분야로 구성된다. 따라서 평가는 이를 네 가지 분야에서의 검증결과를 근거로 이루어지도록 하였다. 소프트웨어의 안전성을 평가받기 위한 업체는 평가기관이 필요로 하는 모든 자료를 제공해야 한다. 평가기관에서는 그 자료와 본 문서를 바탕으로 4 단계로 나누어서 평가한다. 소프트웨어 안전성 등급은 최종 생산물인 소프트웨어 자체의 안전성과 소프트웨어 개발 단계의 완전 무결성을 통합하여 평가된다. 평가자는 소프트웨어 안전성 평가를 위해 다음의 정량적 기준을 이용하여 요구조건들의 만족도를 CMMI(Capability Maturity Model Integration)와 같은 일반적인 소프트웨어 평가단계인 4단계로 평가하도록 다음과 같이 제안하였다.

- F(Fully achieved) 85% 초과 100% 이하 만족
- L(Largely achieved) 50% 초과 85% 이하 만족
- P(Partially achieved) 15% 초과 50% 이하 만족
- N(Not achieved) 0% 이상 15% 이하 만족

그림 6은 소프트웨어 안전성 평가를 위한 가이드라인의 작성 과정을 나타낸 것이며, 문서화 과정의 사례를 제시한 결과이다. IEC61508에서 정의된 소프트웨어 개발의 문서화 요건을 바탕으로 평가 대상 소프트웨어의 문서화 정도를 평가하기 위한 만족도 체크리스트 일부를 나타내고 있다.

**관련표준 → 적용방법설명 → 체크리스트**

(IEC61508-1 5.2.1, 5.2.2) 문서에 필요한 정보가 충분히 기록되었는지 확인한다.  
(IEC61508-1 5.2.5) 문서가 접근 가능인지 확인한다.  
(IEC61508-1 5.2.6, 5.2.7) 문서가 명확하게 작성되었는지 확인한다.  
(IEC61508-1 5.2.8, 5.2.9, 5.2.10, 5.2.11) 문서가 체계적으로 관리되었는지 확인한다.

평가 ID	평가항목	관련 표준	측정 결과	만족도 (N~F)
EC1-EI1	문서에 기록되어야 할 정보가 충분한가?	IEC61508-1 5.2.2		
EC1-EI7	문서의 규격에 일관성이 있는가?	IEC61508-1 5.2.7		

그림 6 소프트웨어 안전성 평가 가이드라인 예

Fig. 6 Example of Software Assessment Guideline

##### 4.2 체크리스트 모듈의 설계

앞 절에서 설명한 열차제어시스템 소프트웨어 평가가이드라인을 관련 국제표준에서 요구하는 기능 안전성 분석을 위한 체크리스트로 활용하였으며, 같은 정적 테스팅 모듈인 Fagan Inspection 모듈과 같이 하나의 웹 시스템으로 설계

하였다. 본 논문에서 구현한 체크리스트 모듈은 체크리스트의 입력·수정·조회·삭제, 체크리스트를 활용한 평가 결과 입력·수정·조회·삭제, 관리자가 체크리스트의 항목들을 관리할 수 있도록 구성하였다. 또한 소프트웨어 안정성 평가 기준에 따라 평가 ID, 평가항목, 관련표준 등을 입력하고 각 항목 별로 수정 및 삭제가 가능하도록 하였고, 항목들이 일정한 기준에 의거하여 큰 변화가 일어날 경우는 체크리스트의 항목들을 묶은 Set 별로 버전 관리가 될 수 있도록 계층별 체크리스트 편집기능을 가지도록 하였다.

이 모듈을 통해 평가자는 관리자가 정의한 체크리스트의 항목을 바탕으로 각 평가 대상들에 대한 평가를 웹 시스템을 이용하여 진행할 수 있어야 하며, 각 항목 별 평가 내용을 입력, 수정, 조회 및 삭제가 가능해야 하고 이는 종합 관리되어 체크리스트 평가 결과물에 대한 전체 평가가 또다시 이루어질 수 있어야 한다. 따라서 체크리스트 모듈도 Fagan Inspection 모듈과 마찬가지로 관리가 관점과 일반 유저 관점에서의 기능의 차이를 두어 이 도구를 이용하여 효과적인 평가가 가능하도록 하였다.

관리자 관점에서의 주요한 기능은 일반 유저 관리 기능(유저 등록, 수정, 삭제, 검색 등), 프로젝트 관리 기능(프로젝트 등록, 수정, 삭제, 프로젝트 관리자 설정, 검색 등) 그리고 새로운 프로젝트 등록 시 프로젝트 관리자에게 메일을 통한 알림 기능으로 구성되며, 체크리스트 버전 관리와 체크리스트 항목 관리 기능도 포함되어 있다. 일반 유저 관점에서는 인스펙션 모듈과 마찬가지로 사용자 개인 정보 관리 기능과 참여 프로젝트 리스트 및 프로젝트 정보 열람 기능을 기본으로 하여 다음과 같은 기능들을 가진다.

- 체크리스트를 활용한 평가 진행, 평가 내용 수정 (항목별 N~F 만족도 평가, 평가에 대한 의견 입력, 수정, 삭제 등)
- 체크리스트 카테고리별 결과 확인
- 체크리스트 진행 현황
- 평가 결과 보고서 작성

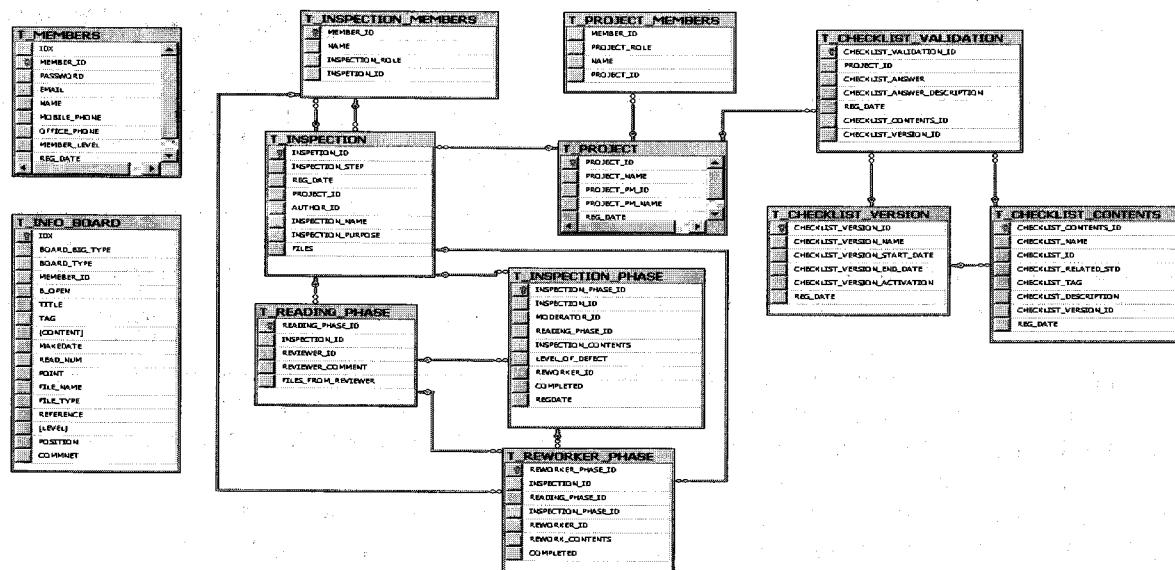


그림 9 데이터베이스 ER 다이어그램

Fig. 9 Database ER Diagram

그림 7과 8은 앞에서 설명한 체크리스트 모듈의 원도우를 설계한 그림으로, 체크리스트의 항목 입력이나 수정 원도우와 체크리스트의 평가 및 수정 원도우를 나타낸 것이다. 이러한

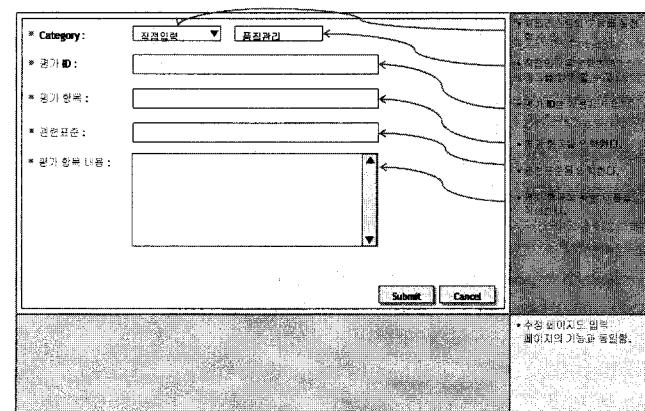


그림 7 체크리스트 항목 입력·수정 원도우

Fig. 7 Inputs and Modification Windows for Checklists

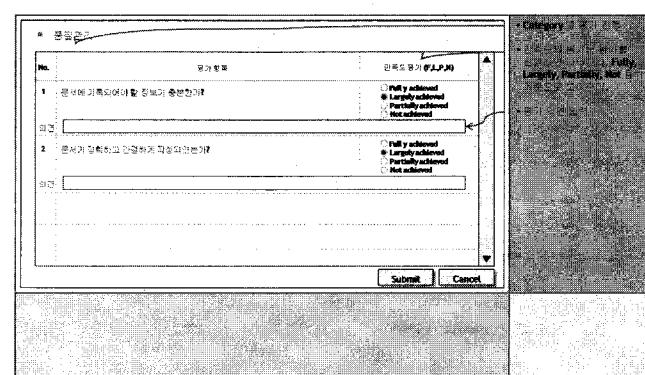


그림 8 체크리스트 평가·수정 원도우

Fig. 8 Assessment and Modification Windows for Checklists

윈도우를 통해 체크리스트를 입력 및 수정하거나, 입력된 리스크를 통해 평가를 입력할 수 있도록 하고 또한 결과들을 레포트 형태로 출력되어 활용할 수 있도록 구현하였다. 앞에서도 설명하였듯이 본 논문에서 제시한 인스펙션 모듈과 체크리스트 모듈은 하나의 웹 시스템으로 구성하였고, 이를 바탕으로 설계한 데이터베이스의 ER Diagram은 그림 9와 같다.

## 5. 정적분석 도구의 개발

앞 장에서 설명한 국제표준 기반의 열차제어시스템 소프트웨어 안전성 평가를 위한 정적 테스팅을 위한 Fagan Inspection 모듈과 체크리스트 모듈을 개발하였다. 그림 10은 Fagan Inspection 모듈의 기능 구성도를 나타낸 것으로서, Fagan Inspection 모듈에서는 인스펙션을 위해 체크리스트와 코딩모듈이 입력되어야 한다. 따라서 체크리스트 모듈과도 인터페이스를 통하여 통합된 하나의 시스템으로 구축하였다. 그림에서와 같이 인스펙션 수행모듈과 인스펙션 결과 모듈 그리고 결과조회 모듈의 세 가지의 하부 모듈을 가지고 있다.

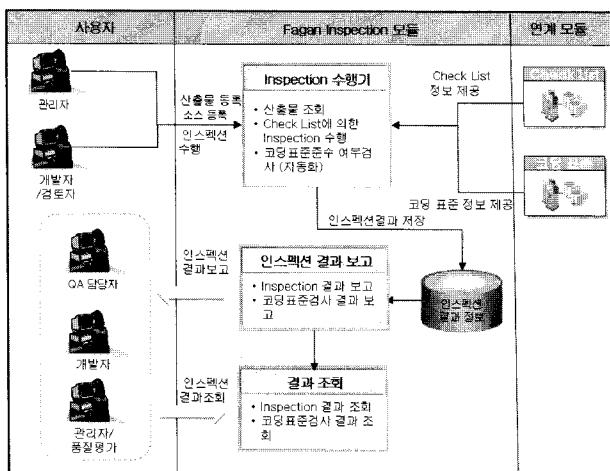


그림 10 Fagan Inspection 모듈의 기능 구성도  
Fig. 10 Function Diagram for Fagan Inspection Module

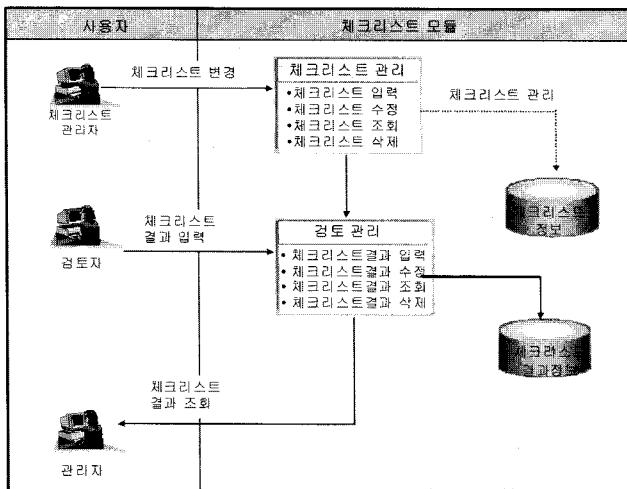


그림 11 체크리스 모듈의 기능 구성도  
Fig. 11 Function Diagram for Checklists Module

그림 11은 체크리스트 모듈의 기능 구성도를 나타낸 것으로, 체크리스트 관리 모듈과 검토관리 모듈이라는 두개의 하부 모듈로 구성되어 있다. 이 체크리스트 모듈을 통해 생성한 체크리스트 모듈은 앞에서 설명한 Fagan Inspection 모듈의 입력으로 사용되어지게 된다.

본 논문을 통해 구현한 정적 테스팅을 위한 지원도구는 MS-SQL Server 2000과 MS-Windows 2003 그리고 MS Internet Explorer 5.0을 기반으로 한 웹 시스템을 구축하였다. 그림 12는 3장에서 설계한 Fagan Inspection 모듈 중 인스펙션 단계의 윈도우 화면을 나타낸 것으로, 이 화면을 통해 Reviewer의 Review 의견 별로 인스펙션을 진행하고 그에 대한 의견을 입력할 수 있다. 또한 해당 인스펙션 내용에 대한 Rework 담당자를 결정할 수 있고 Rework 내용을 다시 인스펙션하여 해당 Review 내용의 완료를 결정할 수 있다.

그림 13은 리워크 단계의 윈도우로서 해당 인스펙션의 내용을 볼 수 있으며, 인스펙션의 개요 및 산출물을 검토한 후 등록한 Reviewer의 Review 의견 열람 및 새로운 Review 의견을 입력할 수 있고, 또한 자신이 쓴 Review 의견뿐만 아니라 해당 인스펙션에 참여하는 다른 Reviewer의 Review

The screenshot shows two windows from the 'KRI 한국철도기술연구원' system. The left window is titled 'Inspection Phase' and contains fields for 'Reviewer' (김철수), 'Review' (리뷰 의견입니다.), 'Plan' (제정한 품질 요구사항이 없습니다.), and '등록일' (2006-07-06 오후 09:34:20). It also includes a note: '본 Review에 대해서 새로운 인스펙션 내용이 있다면 아래에 작성해주시길 바랍니다.' The right window is titled '인스펙션' and shows a report table with columns for '인스펙션' (인스펙션), '오류 수준' (선행화점으로), 'Rework 단장' (선행화점으로), and 'Report' (Report). A status indicator at the bottom says '완료 여부: ○ 예 / ○ 아니오'.

그림 12 인스펙션 단계 윈도우  
Fig. 12 Implemented INSPECTION Phase Windows

The screenshot shows two windows from the 'KRI 한국철도기술연구원' system. The left window is titled 'Rework Phase' and contains fields for 'Reviewer' (김철수), 'Review' (리뷰 의견입니다.), 'Plan' (제정한 품질 요구사항이 없습니다.), and '등록일' (2006-07-06 오후 09:34:20). The right window is titled '인스펙션' and shows a report table with columns for '인스펙션' (인스펙션), '오류 수준' (선행화점으로), 'Rework 단장' (선행화점으로), and 'Report' (Report). A status indicator at the bottom says '완료 여부: ○ 예 / ○ 아니오'.

그림 13 리워크 단계 윈도우  
Fig. 13 Implemented REWORK Phase Windows

의견 역시 함께 검토할 수 있도록 구현하였다. 이러한 일련의 인스펙션 과정들이 구축한 시스템의 데이터베이스에 저장이 되고, 그림 14와 같이 보고서 형태로 출력되도록 하여 모든 인스펙션의 과정과 그 처리결과를 확인할 수 있도록 하였다.

그림 15는 웹 시스템을 통해 평가자가 체크리스트 각 항목별로 평가를 진행하는 화면을 나타낸 것으로 N에서 F까지 판단하는 항목, 적합 또는 부적합으로 판단하는 항목 등으로 구분되어진다. 그림 16은 평가자가 평가한 결과를 나타낸 화면으로 체크리스트 항목별로 평가결과를 전체적으로 확인할 수 있다. 체크리스트 모듈 또한 평가한 결과들이 DB에 저장되고 이를 보고서 형태로 출력 가능하도록 구현하였다.

<b>2 프로젝트 평가 요약 정보</b>																					
프로젝트 평가 요약 정보 프로젝트 평가 요약 정보																					
<table border="1"> <tr><td>프로젝트 명</td><td>테스트 인스펙션</td></tr> <tr><td>체크리스트 목록</td><td>요구사항 분석</td></tr> <tr><td>설명자</td><td>인스펙션 날짜 인스펙션 제시일 승인</td></tr> <tr><td>Review</td><td>테스트 유저1 (test01) 인스펙션 담당 인스펙션 제시일 승인</td></tr> <tr><td>Files</td><td>1. 테스트 문서1.zip &gt;&gt; 다운로드</td></tr> <tr><td>Reviewer</td><td>테스트 유저4 (test04) 첫번째 Review 의견입니다. 이게 이게 잘못됐네요 from test04</td></tr> <tr><td>Files</td><td>1. 새 텍스트 문서3.zip &gt;&gt; 다운로드</td></tr> <tr><td>Inspection</td><td>첫번째 Review 의견에 대한 인스펙션입니다. 이게 이게 잘못됐으나 조치 해주세요</td></tr> <tr><td>오류 수준</td><td>Critical</td></tr> <tr><td>Reviewer</td><td>테스트 유저2 (test02) 이게 이게 잘못된것을 확인했으며</td></tr> </table>		프로젝트 명	테스트 인스펙션	체크리스트 목록	요구사항 분석	설명자	인스펙션 날짜 인스펙션 제시일 승인	Review	테스트 유저1 (test01) 인스펙션 담당 인스펙션 제시일 승인	Files	1. 테스트 문서1.zip >> 다운로드	Reviewer	테스트 유저4 (test04) 첫번째 Review 의견입니다. 이게 이게 잘못됐네요 from test04	Files	1. 새 텍스트 문서3.zip >> 다운로드	Inspection	첫번째 Review 의견에 대한 인스펙션입니다. 이게 이게 잘못됐으나 조치 해주세요	오류 수준	Critical	Reviewer	테스트 유저2 (test02) 이게 이게 잘못된것을 확인했으며
프로젝트 명	테스트 인스펙션																				
체크리스트 목록	요구사항 분석																				
설명자	인스펙션 날짜 인스펙션 제시일 승인																				
Review	테스트 유저1 (test01) 인스펙션 담당 인스펙션 제시일 승인																				
Files	1. 테스트 문서1.zip >> 다운로드																				
Reviewer	테스트 유저4 (test04) 첫번째 Review 의견입니다. 이게 이게 잘못됐네요 from test04																				
Files	1. 새 텍스트 문서3.zip >> 다운로드																				
Inspection	첫번째 Review 의견에 대한 인스펙션입니다. 이게 이게 잘못됐으나 조치 해주세요																				
오류 수준	Critical																				
Reviewer	테스트 유저2 (test02) 이게 이게 잘못된것을 확인했으며																				

그림 14 프로젝트 보고서 보기

Fig. 14 Implemented Windows for Project Report

<b>3 평가 항목 평가</b>																																																									
체크리스트 항목 평가 결과입니다. 평가 항목에 따른 만족도 및 의견을 제공해주세요.																																																									
<table border="1"> <tr><td>평가 항목</td><td>평가 항목</td><td>평가 ID</td><td>관련 표준</td><td>결과</td><td>만족도</td></tr> <tr><td>제작기준의 만족 평가는 기본 만족성을 확보하기 위한 것 디자인 및 기술적 특성들이 규정으로 대응되었는가?</td><td>E02-E01</td><td>IEC61508-15.2.1</td><td>Full Achieved</td><td><input type="radio"/></td></tr> <tr><td>제작기준의 활용과 일상적인 활용 및 기술적 활용의 조건 대 응 구현되었는가?</td><td>E02-E02</td><td>IEC61508-15.2.2</td><td>Partially Achieved</td><td><input type="radio"/></td></tr> <tr><td>제작기준의 전달성을 확보하였는가?</td><td>E02-E03</td><td>IEC61508-15.2.2</td><td>Partially Achieved</td><td><input type="radio"/></td></tr> <tr><td>제작기준의 결과로 인해 성장된 기능을 공식 화되었는가?</td><td>E02-E04</td><td>IEC61508-15.2.3</td><td>Large Achieved</td><td><input type="radio"/></td></tr> <tr><td>기능 안전성 충족에 참여한 자들은 자신의 것 으로 인정하고 있는가?</td><td>E02-E05</td><td>IEC61508-15.2.4</td><td>Full Achieved</td><td><input type="radio"/></td></tr> <tr><td>제작기준을 활용 소프트웨어 개발 단계 전에 제작기준에 맞는 평가를 받았는가?</td><td>E02-E06</td><td>IEC61508-15.2.5</td><td>Not Achieved</td><td><input type="radio"/></td></tr> <tr><td>제작기준에 맞는 평가를 받았는가?</td><td>E02-E07</td><td>IEC61508-15.2.6</td><td>Large Achieved</td><td><input type="radio"/></td></tr> <tr><td>제작기준에 맞는 평가를 받았는가?</td><td>E02-E08</td><td>IEC61508-15.2.6</td><td>Full Achieved</td><td><input type="radio"/></td></tr> <tr><td>제작기준에 맞는 평가를 받았는가?</td><td>E02-E09</td><td>IEC61508-15.2.6</td><td>Partially Achieved</td><td><input type="radio"/></td></tr> <tr><td>제작기준에 맞는 평가를 받았는가?</td><td>E02-E10</td><td>IEC61508-15.2.6</td><td>Large Achieved</td><td><input type="radio"/></td></tr> </table>		평가 항목	평가 항목	평가 ID	관련 표준	결과	만족도	제작기준의 만족 평가는 기본 만족성을 확보하기 위한 것 디자인 및 기술적 특성들이 규정으로 대응되었는가?	E02-E01	IEC61508-15.2.1	Full Achieved	<input type="radio"/>	제작기준의 활용과 일상적인 활용 및 기술적 활용의 조건 대 응 구현되었는가?	E02-E02	IEC61508-15.2.2	Partially Achieved	<input type="radio"/>	제작기준의 전달성을 확보하였는가?	E02-E03	IEC61508-15.2.2	Partially Achieved	<input type="radio"/>	제작기준의 결과로 인해 성장된 기능을 공식 화되었는가?	E02-E04	IEC61508-15.2.3	Large Achieved	<input type="radio"/>	기능 안전성 충족에 참여한 자들은 자신의 것 으로 인정하고 있는가?	E02-E05	IEC61508-15.2.4	Full Achieved	<input type="radio"/>	제작기준을 활용 소프트웨어 개발 단계 전에 제작기준에 맞는 평가를 받았는가?	E02-E06	IEC61508-15.2.5	Not Achieved	<input type="radio"/>	제작기준에 맞는 평가를 받았는가?	E02-E07	IEC61508-15.2.6	Large Achieved	<input type="radio"/>	제작기준에 맞는 평가를 받았는가?	E02-E08	IEC61508-15.2.6	Full Achieved	<input type="radio"/>	제작기준에 맞는 평가를 받았는가?	E02-E09	IEC61508-15.2.6	Partially Achieved	<input type="radio"/>	제작기준에 맞는 평가를 받았는가?	E02-E10	IEC61508-15.2.6	Large Achieved	<input type="radio"/>
평가 항목	평가 항목	평가 ID	관련 표준	결과	만족도																																																				
제작기준의 만족 평가는 기본 만족성을 확보하기 위한 것 디자인 및 기술적 특성들이 규정으로 대응되었는가?	E02-E01	IEC61508-15.2.1	Full Achieved	<input type="radio"/>																																																					
제작기준의 활용과 일상적인 활용 및 기술적 활용의 조건 대 응 구현되었는가?	E02-E02	IEC61508-15.2.2	Partially Achieved	<input type="radio"/>																																																					
제작기준의 전달성을 확보하였는가?	E02-E03	IEC61508-15.2.2	Partially Achieved	<input type="radio"/>																																																					
제작기준의 결과로 인해 성장된 기능을 공식 화되었는가?	E02-E04	IEC61508-15.2.3	Large Achieved	<input type="radio"/>																																																					
기능 안전성 충족에 참여한 자들은 자신의 것 으로 인정하고 있는가?	E02-E05	IEC61508-15.2.4	Full Achieved	<input type="radio"/>																																																					
제작기준을 활용 소프트웨어 개발 단계 전에 제작기준에 맞는 평가를 받았는가?	E02-E06	IEC61508-15.2.5	Not Achieved	<input type="radio"/>																																																					
제작기준에 맞는 평가를 받았는가?	E02-E07	IEC61508-15.2.6	Large Achieved	<input type="radio"/>																																																					
제작기준에 맞는 평가를 받았는가?	E02-E08	IEC61508-15.2.6	Full Achieved	<input type="radio"/>																																																					
제작기준에 맞는 평가를 받았는가?	E02-E09	IEC61508-15.2.6	Partially Achieved	<input type="radio"/>																																																					
제작기준에 맞는 평가를 받았는가?	E02-E10	IEC61508-15.2.6	Large Achieved	<input type="radio"/>																																																					

그림 15 Checklist 항목 평가 진행

Fig. 15 Implemented Windows for Assessing Checklists

## 6. 결 론

최근 컴퓨터 기술의 발달에 따라 열차제어시스템들이 컴퓨터 소프트웨어에의 의존성이 급격하게 증가하고 있으며, 이러한 기술발전에 따라 열차제어시스템 소프트웨어에 높은 신뢰성과 안전성이 요구되고 있다.

<b>4 체크리스트 항목 평가</b>																																																								
체크리스트 항목 평가 결과입니다.																																																								
<table border="1"> <tr><td>평가 항목</td><td>평가 ID</td><td>관련 표준</td><td>결과</td><td>만족도</td></tr> <tr><td>제작기준의 활용과 일상적인 활용의 조건 대 응 구현되었는가?</td><td>E02-E01</td><td>IEC61508-15.2.1</td><td>Large Achieved</td><td><input type="radio"/></td></tr> <tr><td>제작기준의 결과로 인해 성장된 기능을 공식 화되었는가?</td><td>E02-E02</td><td>IEC61508-15.2.2</td><td>Partially Achieved</td><td><input type="radio"/></td></tr> <tr><td>제작기준의 전달성을 확보하였는가?</td><td>E02-E03</td><td>IEC61508-15.2.2</td><td>Partially Achieved</td><td><input type="radio"/></td></tr> <tr><td>기능 안전성 충족에 참여한 자들은 자신의 것 으로 인정하고 있는가?</td><td>E02-E04</td><td>IEC61508-15.2.3</td><td>Large Achieved</td><td><input type="radio"/></td></tr> <tr><td>제작기준을 활용 소프트웨어 개발 단계 전에 제작기준에 맞는 평가를 받았는가?</td><td>E02-E05</td><td>IEC61508-15.2.5</td><td>Not Achieved</td><td><input type="radio"/></td></tr> <tr><td>제작기준에 맞는 평가를 받았는가?</td><td>E02-E06</td><td>IEC61508-15.2.6</td><td>Large Achieved</td><td><input type="radio"/></td></tr> <tr><td>제작기준에 맞는 평가를 받았는가?</td><td>E02-E07</td><td>IEC61508-15.2.6</td><td>Full Achieved</td><td><input type="radio"/></td></tr> <tr><td>제작기준에 맞는 평가를 받았는가?</td><td>E02-E08</td><td>IEC61508-15.2.6</td><td>Partially Achieved</td><td><input type="radio"/></td></tr> <tr><td>제작기준에 맞는 평가를 받았는가?</td><td>E02-E09</td><td>IEC61508-15.2.6</td><td>Large Achieved</td><td><input type="radio"/></td></tr> <tr><td>제작기준에 맞는 평가를 받았는가?</td><td>E02-E10</td><td>IEC61508-15.2.6</td><td>Large Achieved</td><td><input type="radio"/></td></tr> </table>		평가 항목	평가 ID	관련 표준	결과	만족도	제작기준의 활용과 일상적인 활용의 조건 대 응 구현되었는가?	E02-E01	IEC61508-15.2.1	Large Achieved	<input type="radio"/>	제작기준의 결과로 인해 성장된 기능을 공식 화되었는가?	E02-E02	IEC61508-15.2.2	Partially Achieved	<input type="radio"/>	제작기준의 전달성을 확보하였는가?	E02-E03	IEC61508-15.2.2	Partially Achieved	<input type="radio"/>	기능 안전성 충족에 참여한 자들은 자신의 것 으로 인정하고 있는가?	E02-E04	IEC61508-15.2.3	Large Achieved	<input type="radio"/>	제작기준을 활용 소프트웨어 개발 단계 전에 제작기준에 맞는 평가를 받았는가?	E02-E05	IEC61508-15.2.5	Not Achieved	<input type="radio"/>	제작기준에 맞는 평가를 받았는가?	E02-E06	IEC61508-15.2.6	Large Achieved	<input type="radio"/>	제작기준에 맞는 평가를 받았는가?	E02-E07	IEC61508-15.2.6	Full Achieved	<input type="radio"/>	제작기준에 맞는 평가를 받았는가?	E02-E08	IEC61508-15.2.6	Partially Achieved	<input type="radio"/>	제작기준에 맞는 평가를 받았는가?	E02-E09	IEC61508-15.2.6	Large Achieved	<input type="radio"/>	제작기준에 맞는 평가를 받았는가?	E02-E10	IEC61508-15.2.6	Large Achieved	<input type="radio"/>
평가 항목	평가 ID	관련 표준	결과	만족도																																																				
제작기준의 활용과 일상적인 활용의 조건 대 응 구현되었는가?	E02-E01	IEC61508-15.2.1	Large Achieved	<input type="radio"/>																																																				
제작기준의 결과로 인해 성장된 기능을 공식 화되었는가?	E02-E02	IEC61508-15.2.2	Partially Achieved	<input type="radio"/>																																																				
제작기준의 전달성을 확보하였는가?	E02-E03	IEC61508-15.2.2	Partially Achieved	<input type="radio"/>																																																				
기능 안전성 충족에 참여한 자들은 자신의 것 으로 인정하고 있는가?	E02-E04	IEC61508-15.2.3	Large Achieved	<input type="radio"/>																																																				
제작기준을 활용 소프트웨어 개발 단계 전에 제작기준에 맞는 평가를 받았는가?	E02-E05	IEC61508-15.2.5	Not Achieved	<input type="radio"/>																																																				
제작기준에 맞는 평가를 받았는가?	E02-E06	IEC61508-15.2.6	Large Achieved	<input type="radio"/>																																																				
제작기준에 맞는 평가를 받았는가?	E02-E07	IEC61508-15.2.6	Full Achieved	<input type="radio"/>																																																				
제작기준에 맞는 평가를 받았는가?	E02-E08	IEC61508-15.2.6	Partially Achieved	<input type="radio"/>																																																				
제작기준에 맞는 평가를 받았는가?	E02-E09	IEC61508-15.2.6	Large Achieved	<input type="radio"/>																																																				
제작기준에 맞는 평가를 받았는가?	E02-E10	IEC61508-15.2.6	Large Achieved	<input type="radio"/>																																																				

KRI 한국철도기술연구원

Copyright © 2009 Korea Railroad Research Institute. All Rights Reserved.

리스트

그림 16 Checklist 항목 평가 결과

Fig. 16 Implemented Windows for Assessed Checklists

이에 따라 바이탈 세이징장치인 열차제어시스템의 소프트웨어 안전성 평가를 위해서는 관련된 국제규격에서 소프트웨어의 분석 및 측정을 통한 정량적인 평가뿐만 아니라 소프트웨어 개발과정의 안전성 활동에 대한 문서 등의 검증을 통한 정적 테스팅도 안전성 평가의 중요한 요소로 요구하고 있다. 본 논문에서는 관련 국제 표준에서 정적 테스팅의 방법으로 요구하고 있는 Fagan Inspection 및 체크리스트 평가를 지원하기 위한 웹 기반의 도구를 개발하였다. 이러한 정적 테스팅을 위한 웹 기반의 지원 도구는 소프트웨어의 안전성 평가 단계에서 활동될 도구이며, 동시에 소프트웨어 개발과정에서 활용되어질 수도 있도록 화면을 구성하는 등 다양하게 활용될 수 있도록 개발하였다. 즉, 본 도구를 소프트웨어 수명주기 각 단계별 인스펙션에 활용할 수 있고, 이를 통해 보다 높은 안전성과 신뢰성을 갖는 소프트웨어가 확보될 수 있을 것으로 기대된다.

## 감사의 글

본 논문은 국토해양부가 출연하고 한국건설교통기술평가원에서 위탁 시행한 철도종합안전기술개발사업의 결과이며, 관계제위께 감사드립니다.

## 참 고 문 헌

- [1] IEC 61508, "Railway Applications - The specification and demonstration of RAMS", 1998.

- [2] IEC 62279, "Railway Applications – Software for railway control and protection systems", 2002.
- [3] 철도안전법[법률 8852호], 일부개정 2008. 02.
- [4] N. Eickelmann, F. Ruffolo, etc., "An empirical study of modifying the Fagan Inspection process and the resulting main effects and interaction effects among defects found, effort required, rate of preparation and inspection", Proceedings of the 27th Annual NASA Goddard Software Engineering Workshop, 2002.
- [5] 소선섭, 차성덕, 권용래, T.J.Shimeall, "페이션 인스펙션의 오류 검출 능력에 관한 실험적 평가", 정보과학회 논문지, 제24권 제12호, 1997. 12.
- [6] Ian B. Pirie, "Software - How do we know it is safe?", Proceedings of the 1999 ASME/IEEE Joint, pp.122-129, 1999.
- [7] J.D. Lawrence, "Software qualification in safety applications", Reliability Engineering & System Safety, Vol. 70, No. 2, pp. 167-184, 2000.
- [8] 황종규, 조현정, 김형신, "열차제어시스템 소프트웨어 안전성 평가도구의 설계", 한국철도학회 논문집, 제11권 제2호, pp. 139-144, 2008. 4.

## 저자 소개



황종규 (黃宗奎)

1994년 건국대학교 전기공학과 졸업.  
1996년 동 대학원 석사졸업. 2005년 한양  
대학교 전자통신전파공학과 박사졸업.  
1995년~현재 한국철도기술연구원 열차  
제어통신연구실 선임연구원.

Tel : 031-460-5438

Fax : 031-460-5449

E-mail : jghwang@krri.re.kr



조현정 (趙賢庭)

2003년 한국항공대학교 항공전자공학과  
졸업. 2005년 광주과학기술원(GIST) 정  
보통신공학과 졸업. 2005년~현재 한국철  
도기술연구원 열차제어통신연구실 주임  
연구원.

Tel : 031-460-5458

Fax : 031-460-5449

E-mail : hjo@krri.re.kr